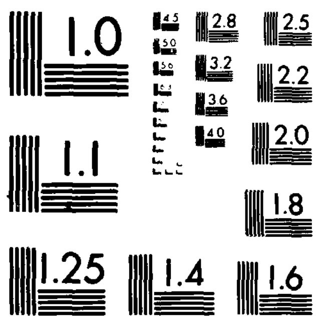
AD-A108 357 NAVAL RESEARCH LAB WASHINGTON DC F/G 12/1
ALGORITHM FOR THE INVERSE OF A HERMITIAN TOEPILTZ MATRIX.(U)
NOV 81 K GERLACH
UNCLASSIFIED NRL-8539 NL

END
DATE
FILMED
82
DTIC



MICROCOPY RESOLUTION TEST CHART

NATIONAL BUREAU OF STANDARDS-1963-A

LEVEL II

NRL Report 8539

AD A108357

Algorithm for the Inverse of a Hermitian Toepiltz Matrix

KARL GERLACH

Airborne Radar Branch
Radar Division

November 27, 1981

NAVAL RESEARCH LABORATORY
Washington, D.C.

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER NRL Report 8539	2. GOVT ACCESSION NO. AD-A108 357	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) ALGORITHM FOR THE INVERSE OF A HERMITIAN TOEPILTZ MATRIX		5. TYPE OF REPORT & PERIOD COVERED Interim report on a continuing NRL problem. 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Karl Gerlach		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Research Laboratory Washington, DC 20375		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS Program element 62712N 53-0662-0-1 WF12-000-001
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Air Systems Command Washington, DC 20361		12. REPORT DATE November 27, 1981 13. NUMBER OF PAGES 17
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Matrix Inverse matrix Toepiltz Hermitian		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) New methods for finding the inverse of a hermitian Toepiltz matrix and the related problem of solving a system of linear equations are presented. Efficient algorithmic procedures for these methods are listed.		

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

251950

CONTENTS

ALGORITHM FOR THE INVERSE OF A HERMITIAN TOEPILTZ MATRIX

INTRODUCTION

The efficient inversion of a given matrix and the related problem of solving a system of linear equations has been a subject of intense study for many years. The literature on this subject is so vast that no survey can be exhaustive. For example, a tentative classification and bibliography on solving systems of linear equations written by Forsythe [1] contains over 400 titles. An excellent handbook on the various numerical methods of matrix inversion and the solution of linear equations has been written by Westlake [2]. Different methods are compared based on such measures of effectiveness as speed, storage requirements, and convergence rates if applicable.

Numerical methods for matrix inversion and the related problem of solving a system of linear equations can be divided into two classes: the direct methods and the indirect (iterative) methods. Direct methods such as Cramer's rule [3], Gaussian elimination [3], and orthogonalization [3-4] yield an exact solution after a finite number of operations if there is no roundoff error. Iterative methods on the other hand such as gradient methods [4], the back and forth Seidel [4], and successive overrelation [5], begin with an approximate solution and obtain an improved solution with each step of the iteration. The accuracy of the solution depends on the number of iterations performed.

For most direct methods of matrix inversion, the number of arithmetic operations is proportional to M^3 where M is the row or column dimension of the given square matrix. For iterative methods, the number of operations per iteration is proportional to M^2. In general, the speed of an algorithm if there is no parallel processing is proportional to the number of arithmetic operations so that this measure can be used to evaluate the performance of a given algorithm.

A direct procedure for finding the solution of simultaneous linear equations where the multiplying matrix is Toepiltz was developed by Levinson and presented in Norbert Weiner's book, *Extrapolation, Interpolation, and Smoothing of Stationary Time Series* [6]. This algorithm takes advantage of the Toepiltz form to reduce the number of arithmetic operations to be proportional to M^2. This algorithm has been used by Burg [7] to estimate line spectra in a methodology commonly called maximum entropy spectrum analysis (MESA).

This report presents a new direct method for finding the inverse of an $M \times M$ hermitian Toepiltz matrix. An $M \times M$ hermitian Toepiltz matrix, H, has the form

$$H = \begin{bmatrix} h_0 & h_1^* & h_2^* & \cdots & h_{M-1}^* \\ h_1 & h_0 & h_1^* & \cdots & h_{M-2}^* \\ h_2 & h_1 & h_0 & \cdots & h_{M-3}^* \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ h_{M-1} & h_{M-2} & h_{M-3} & \cdots & h_0 \end{bmatrix} \tag{1}$$

Manuscript submitted August 16, 1981.

where h_0 is always real and * indicates the complex conjugate. Note that it is only necessary to specify the elements of the first column of a hermitian Toepiltz matrix in order to define the entire matrix. Therefore, we introduce the shortened notation: if H is an $M \times M$ hermitian Toepiltz matrix, then we write

$$H = ((h_0, h_1, h_2, \ldots, h_{M-1})) \tag{2}$$

where h_k, $k = 0, 1, \ldots, M - 1$ are the elements of the first column of H.

We will take advantage of the form of a hermitian Toepiltz matrix and develop new direct methods for the solution of simultaneous linear equations and the matrix inverse. The basis of these related algorithms lies in discrete Fouries series theory. Efficient algorithmic procedures are presented which use the theory of the preceding sections to find the matrix inverse and the solution of simultaneous linear equations respectively. We then discuss the software implementation of the matrix inversion algorithm.

SOLUTION OF SIMULTANEOUS LINEAR EQUATIONS

In this section, we will develop an algorithm for solving for the unknowns of a system of M independent linear equations. Using this algorithm, we will see in the next section that an algorithm for obtaining the inverse of a given hermitian Toepiltz matrix can be derived.

Consider the vector equation

$$H\bar{x} = \bar{c} \tag{3}$$

where H is an $M \times M$ nonsingular hermitian Toepiltz matrix, $\bar{c}$ is an $M \times 1$ known vector, and $\bar{x}$ is an $M \times 1$ unknown vector. We desire to find $\bar{x}$. We use the following approach. Let us define a system of $N = 2M - 1$ independent linear equations as

$$\left[\begin{array}{c|c} P_{11}^{(M)} & P_{12}^{(M)} \\ \hline P_{21}^{(M)} & P_{22}^{(M)} \end{array}\right] \begin{bmatrix} \bar{x}_M \\ \bar{0} \end{bmatrix} = \begin{bmatrix} \bar{c}_M \\ \bar{x}_{M-1} \end{bmatrix} \tag{4}$$

where $\bar{x}_M$ is an $M \times 1$ unknown vector, $\bar{c}_M = \bar{c}$, $\bar{x}_{M-1}$ is an $(M - 1) \times 1$ unknown vector, $\bar{0}$ is a zero filled $(M - 1) \times 1$ vector, and

$$\left[\begin{array}{c|c} P_{11}^{(M)} & P_{12}^{(M)} \\ \hline P_{21}^{(M)} & P_{22}^{(M)} \end{array}\right] = ((h_0, h_1, \ldots, h_{M-1}, h_{M-1}^*, h_{M-2}^*, \ldots, h_1^*)) \tag{5}$$

$$\triangleq P_M.$$

We call a matrix defined by the form seen in Eq. (5) as an up-down hermitian Toepiltz matrix (UDHTM) because the subscripts of h_k seen in Eq. (5) increase and then decrease. We also assume that P_M is nonsingular. The matrix, P_M, as seen in Eqs. (4) and (5) is partitioned as follows: $P_{11}^{(M)}$ is a $M \times M$ matrix, $P_{12}^{(M)}$ is an $M \times (M - 1)$ matrix, $P_{21}^{(M)}$ is an $(M - 1) \times M$ matrix, and $P_{22}^{(M)}$ is an $(M - 1) \times (M - 1)$ matrix. In addition, we can also show that $P_{11}^{(M)} = H$ and thus is hermitian Toepiltz and that $P_{22}^{(M)}$ is also hermitian Toepiltz. In fact

$$P_{22}^{(M)} = ((h_0, h_1, h_2, \ldots, h_{M-2})). \tag{6}$$

Note that the system of equations defined by Eq. (4) contains $(2M - 1)$ unknowns and has a unique solution if $P_{11}^{(M)}$ is nonsingular. Also note from Eq. (4) that if $P_{11}^{(M)} = H$ and $\bar{c}_M = \bar{c}$, then $\bar{x}_M = \bar{x}$. Thus, if we solve for the unknowns in Eq. (4), we have also solved for $\bar{x}$ in Eq. (3).

Let us rewrite Eq. (4) as

$$\begin{bmatrix} \bar{x}_M \\ 0 \end{bmatrix} = \left[\begin{array}{c|c} Q_{11}^{(M)} & Q_{12}^{(M)} \\ \hline Q_{21}^{(M)} & Q_{22}^{(M)} \end{array}\right] \begin{bmatrix} \bar{c} \\ \bar{x}_{M-1} \end{bmatrix} \tag{7}$$

where

$$\left[\begin{array}{c|c} Q_{11}^{(M)} & Q_{12}^{(M)} \\ \hline Q_{21}^{(M)} & Q_{22}^{(M)} \end{array}\right] = P_M^{-1} \triangleq Q_M \tag{8}$$

such that $Q_{11}^{(M)}$ is an $M \times M$ matrix, $Q_{12}^{(M)}$ is an $M \times (M-1)$ matrix, $Q_{21}^{(M)}$ is an $(M-1) \times M$ matrix, and $Q_{22}^{(M)}$ is an $(M-1) \times (M-1)$ matrix. We show in Appendix A that

$$Q_M = \frac{1}{N} F_N^* \Lambda^{-1} F_N \tag{9}$$

where F_N is the Nth order discrete Fourier series (DFS) matrix defined by Eq. (A6), and Λ is a diagonal matrix whose element, λ_{kk}, consists of the Nth order DFS of the sequence $\{h_0, h_1, \ldots, h_{M-1}, h_{M-1}^*, h_{M-2}^*, \ldots, h_1^*\}$ (the Nth order DFS is defined by Eq. A4). In fact, if

$$\{s_0, s_1, \ldots, s_{N-1}\} = \text{DFS}\,\{h_0, h_1, \ldots, h_{M-1}, h_{M-1}^*, \ldots, h_1^*\} \tag{10}$$

where $\{s_0, s_1, \ldots, s_{N-1}\}$ is the sequence that results by finding the DFS of the sequence $\{h_0, h_1, \ldots, h_{M-1}, h_{M-1}^*, \ldots, h_1^*\}$, then

$$\lambda_{kk} = s_{k-1}; \quad k = 1, 2, \ldots, N. \tag{11}$$

It is also shown in Appendix A that if P_M is a UDHTM then P_M^{-1} or Q_M is also a UDHTM. Thus we see that Q_M can be written

$$Q_M = ((g_0, g_1, \ldots g_{M-1}, g_{M-1}^*, \ldots, g_1^*)). \tag{12}$$

Hence, it is seen that $Q_{11}^{(M)}$ and $Q_{22}^{(M)}$ are hermitian Toepiltz matrices with

$$Q_{11}^{(M)} = ((g_0, g_1, \ldots, g_{M-1})) \tag{13a}$$

and

$$Q_{22}^{(M)} = ((g_0, g_1, \ldots, g_{M-2})). \tag{13b}$$

Now, let us rewrite Eq. (7) in the equivalent form as

$$\bar{x}_M = Q_{11}^{(M)} \bar{c}_M + Q_{12}^{(M)} \bar{x}_{M-1} \tag{14a}$$

$$\bar{0} = Q_{21}^{(M)} \bar{c}_M + Q_{22}^{(M)} \bar{x}_{M-1}. \tag{14b}$$

Equation (14b) can be rewritten as

$$P_{11}^{(M-1)}\, \bar{x}_{M-1} = \bar{c}_{M-1} \tag{14c}$$

where we have defined

$$P_{11}^{(M-1)} = Q_{22}^{(M)};\ \bar{c}_{M-1} = -\,Q_{21}^{(M)}\, \bar{c}_M. \tag{15}$$

If we had a solution for $\bar{x}_{M-1}$ in Eq. (14a), we could find $\bar{x}_M$. To find $\bar{x}_{M-1}$, we use Eq. (14c). However, $P_{11}^{(M-1)}$ is a $(M-1) \times (M-1)$ hermitian Toepiltz matrix and $\bar{c}_{M-1}$ is a derivable $(M-1) \times 1$ vector. Thus, we have reduced the order of the problem from finding an unknown $M \times 1$ vector, $\bar{x}_M$, to finding an unknown $(M-1) \times 1$ vector, $\bar{x}_{M-1}$, whose multiplying matrix is also hermitian Toepiltz. Hence, the above procedure is reiterative and must be repeated $M - 1$ times with the assumption that $P_{11}^{(k)}$ $k = 2, \ldots, M$ are nonsingular. On the $M - 1$ iteration, the equations have the form

$$\bar{x}_2 = Q_{11}^{(2)}\, \bar{c}_2 + Q_{12}^{(2)}\, \bar{x}_1 \tag{16a}$$

$$\bar{x}_1 = \frac{-\,Q_{21}^{(2)}\, \bar{c}_2}{Q_{22}^{(2)}}. \tag{16b}$$

Note that $\bar{x}_1$ and $Q_{22}^{(2)}$ are now scalars. Thus, no matrix inversion of $Q_{22}^{(2)}$ is necessary (it is assumed $Q_{22}^{(2)} \neq 0$). Therefore, $\bar{x}_1$ is known and $\bar{x}_2$ (a 2×1 unknown vector) can be found by using Eq. (16a). In general, the unknowns, $\bar{x}_k$, $k = 2, 3, \ldots, M$ can be obtained by using the forward reiterative formula

$$\bar{x}_k = Q_{11}^{(k)}\, \bar{c}_k + Q_{12}^{(k)}\, \bar{x}_{k-1}. \tag{17}$$

The constant $k \times 1$ vector, $\bar{c}_k$, $k = 2, 3, \ldots, M$ can be obtained by using the backward reiterative formula

$$\bar{c}_{k-1} = -\,Q_{21}^{(k)}\, \bar{c}_k;\ k = M, M-1, \ldots, 2 \tag{18}$$

with the final condition that $\bar{c}_M = \bar{c}$. In the discussion of software algorithm for matrix inversion, we discuss how to obtain the matrix, Q_k, $k = M, M-1, \ldots, 2$.

TOEPILTZ MATRIX INVERSION ALGORITHM

We can use the algorithm for finding the unknowns of a system of linear equations discussed in the preceding section to obtain the inverse of a given hermitian Toepiltz matrix. Let us define the $k \times k$ matrix, Ω_k, such that

$$\Omega_k \triangleq [P_{11}^{(k)}]^{-1};\ k = 2, 3, \ldots, M \tag{19}$$

and

$$\Omega_1 = \frac{1}{Q_{22}^{(2)}}. \tag{20}$$

Note from Eqs. (3), (4), and (19) that $\Omega_M = H^{-1}$. Now for $k = M$, Eq. (14c) implies that

$$\bar{x}_M = \Omega_M\, \bar{c}_M. \tag{21}$$

Equations (21) and (15) imply that

$$\bar{x}_{M-1} = \Omega_{M-1}\bar{c}_{M-1} = -\,\Omega_{M-1}\, Q_{21}^{(M)}\, \bar{c}_M = -\,\Omega_{M-1}\, Q_{M-1}\, Q_{21}^{(M)}\, \bar{c}. \tag{22}$$

Therefore, if we substitute Eqs. (21) and (22) into Eq. (17), we obtain

$$\begin{aligned}\Omega_M \bar{c} &= Q_{11}^{(M)} \bar{c} - Q_{12}^{(M)} \Omega_{M-1} Q_{21}^{(M)} \bar{c} \\ &= (Q_{11}^{(M)} - Q_{12}^{(M)} \Omega_{M-1} Q_{21}^{(M)}) \bar{c}.\end{aligned} \tag{23}$$

Because $\bar{c}$ is an arbitrary $M \times 1$ vector, Eq. (23) implies the following formula:

$$\Omega_M = Q_{11}^{(M)} - Q_{12}^{(M)} \Omega_{M-1} Q_{21}^{(M)}. \tag{24a}$$

Similarly, we can find a formula for $\Omega_M = [P_{11}^{(M-1)}]^{-1}$. We do this by choosing a new arbitrary vector, $\bar{c}$, of length $M-1$, and initiate solving a system of $M-1$ simultaneous equations as we did in the preceding section. Using equations similar to Eqs. (21) to (23), we would derive an equation exactly like Eq. (24a) except that the index is $M-1$. Hence it is possible to write a reiterative formula

$$\Omega_k = Q_{11}^{(k)} - Q_{12}^{(k)} \Omega_{k-1} Q_{21}^{(k)} \tag{24b}$$

with $k = 2, 3, \ldots, M$, and with Ω_1 given by Eq. (20). Thus if we reiterate Eq. (24b) $M - 1$ times, we obtain $H^{-1} = \Omega_M$.

SOFTWARE ALGORITHM FOR MATRIX INVERSION

In this section, we present an efficient procedure for obtaining the inverse of a hermitian Toepiltz by using the methodology described in the preceding sections. To begin with, it is seen from Eqs. (20) and (24b) that all that is necessary for computing the Ω_k matrices, $k = 1, 2, \ldots, M$ are the Q_k matrices. The partitions, $Q_{11}^{(k)}$, $Q_{12}^{(k)}$, $Q_{21}^{(k)}$, and $Q_{22}^{(k)}$ can be obtained easily from Q_k. Now Q_k is a UDHTM, so that all that is necessary to completely specify it is the first column of the matrix (actually because of the up-down property, just the first M elements of the first column are needed). The matrix Q_k can be found by using the formula

$$Q_k = \frac{1}{2k-1} F_{2k-1}^{*} \Lambda_k^{-1} F_{2k-1}; \quad k = 2, 3, \ldots, M \tag{25}$$

where F_{2k-1} is the $(2k-1)$ order DFS matrix defined by Eq. (A6) and Λ_k is a diagonal matrix. The diagonal element, $\lambda_{ll}^{(k)}$, $l = 1, 2, \ldots, 2k-1$ is found as follows. If

$$Q_{k+1} = ((g_0^{(k+1)}, g_1^{(k+1)}, \ldots, g_k^{(k+1)}, g_k^{(k+1)*}, \ldots, g_1^{(k+1)*})) \tag{26}$$

and

$$\{s_0^{(k)}, s_1^{(k)}, \ldots, s_{2k-2}^{(k)}\} = \text{DFS}\,\{g_0^{(k+1)}, g_1^{(k+1)}, \ldots, g_{k-1}^{(k+1)}, g_{k-1}^{(k+1)*}, \ldots, g_1^{(k+1)*}\}, \tag{27}$$

then

$$\lambda_{ll}^{(k)} = s_{l-1}^{(k)}; \quad l = 1, 2, \ldots, 2k-1. \tag{28}$$

Now in order to generate all Q_k, Λ_M must be known. However, the matrix, Λ_M, can be obtained by using the elements that define H and Eqs. (10) and (11).

To evaluate Q_k, seen in Eq. (25), it is not necessary to perform all of the matrix operations indicated by this equation. In fact, it is straight forward to show (see Appendix A) that

$$\{g_0^{(k)}, g_1^{(k)*}, \ldots, g_{k-1}^{(k)*}, g_{k-1}^{(k)}, \ldots, g_1^{(k)}\} = \text{DFS}\left\{\frac{1}{(2k-1)\lambda_{11}^{(k)}}, \frac{1}{(2k-1)\lambda_{22}^{(k)}}, \ldots, \frac{1}{(2k-1)\lambda_{(2k-1)(2k-1)}^{(k)}}\right\}. \tag{29}$$

Thus, based upon the preceding discussion, we present the following algorithmic procedure for finding H^{-1}:

A. Set $k = M - 1$, $g_l^{(M+1)} = h_l$, $l = 0, 1, \ldots, M - 1$.

B. Calculate $\{s_0^{(k)}, s_1^{(k)}, \ldots, s_{2k-1}^{(k)}\}$ by using Eq. (27).

C. Calculate $\{g_0^{(k)}, g_1^{(k)}, \ldots, g_{k-1}^{(k)}\}$ by using Eqs. (28) and (29).

D. Store $\{g_0^{(k)}, g_1^{(k)}, \ldots, g_{k-1}^{(k)}\}$.

E. $k = k - 1$.

F. Go to B if $k > 1$.

G. Set $\Omega_1 = \dfrac{1}{g_0^{(2)}}$ (note $g_0^{(2)} = Q_{22}^{(2)}$) and $k = 2$.

H. Set $Q_k = ((g_0^{(k)}, g_1^{(k)}, \ldots, g_{k-1}^{(k)}, g_{k-1}^{(k)*}, \ldots, g_1^{(k)*}))$.

I. Construct partitions: $Q_{11}^{(k)}, Q_{12}^{(k)}, Q_{21}^{(k)}$.

J. $\Omega_k = Q_{11}^{(k)} - Q_{12}^{(k)} \Omega_{k-1} \Omega_{21}^{(k)}$.

K. $k = k + 1$.

L. Go to H if $k \leqslant M$.

M. $H^{-1} = \Omega_M$.

The algorithm can be divided into two parts: the first part (steps A-F) consists of finding the elements of Q_k, $k = 2, 3, \ldots, M$, and the second part (steps G-M) calculates through a reiterative formula (step J) the Ω_k matrices.

It can be shown that $g_0^{(k)}$ and $s_l^{(k)}$, $l = 0, 1, \ldots, 2k - 2$ are always real. Because of computational errors, however, these values may have a small imaginary part. It was found that the accuracy of the matrix inverse, H^{-1}, improved if only the real part of the computed $g_0^{(k)}$ or $s_l^{(k)}$ was used in succeeding steps of the algorithm.

A Fortran computer program listing that implements the matrix inversion algorithm is given in Appendix B.

SOFTWARE ALGORITHM FOR SOLUTION OF SIMULTANEOUS EQUATIONS

Similar to the preceding section, we present an algorithmic procedure for finding the solution of a system of simultaneous linear equations as given by Eq. (3) as follows:

A. Set $k = M$, $g_l^{(M+1)} = h_l$; $l = 0, 1, 2, \ldots, M - 1$, $\bar{c}_M = \bar{c}$.

B. Calculate $\{s_0^{(k)}, s_1^{(k)}, \ldots, s_{2k-1}^{(k)}\}$ by using Eq. (27).

C. Calculate $\{g_0^{(k)}, g_1^{(k)}, \ldots, g_{k-1}^{(k)}\}$ by using Eqs. (28) and (29).

D. Store $\{g_0^{(k)}, g_1^{(k)}, \ldots, g_{k-1}^{(k)}\}$.

E. Construct partition $Q_{21}^{(k)}$ by using $\{g_0^{(k)}, g_1^{(k)}, \ldots, g_{k-1}^{(k)}\}$.

F. $\bar{c}_{k-1} = -Q_{21}^{(k)}\bar{c}_k$; store $\bar{c}_{k-1}$.

G. $k = k - 1$.

H. Go to B if $k > 1$.

I. Set $\bar{x}_1 = \bar{c}_1/g_0^{(2)}$ (note $\bar{x}_1$, $\bar{c}_1$ are scalars) and $k = 2$.

J. Set $Q_k = ((g_0^{(k)}, g_1^{(k)}, \ldots, g_{k-1}^{(k)}, g_{k-1}^{(k)*}, \ldots, g_1^{(k)*}))$.

K. Construct partitions: $Q_{11}^{(k)}$, $Q_{12}^{(k)}$.

L. $\bar{x}_k = Q_{11}^{(k)}\bar{c}_k + Q_{12}^{(k)}\bar{x}_{k-1}$.

M. $k = k + 1$.

N. Go to J if $k \leqslant M$.

O. $\bar{x} = \bar{x}_M$.

IMPLEMENTATION OF THE MATRIX INVERSION ALGORITHM

The value of any algorithm that is used as a computer library subroutine is determined by such measures as speed, the amount of computer memory needed, and the amount of hardware necessary to implement the algorithm. The last two measures can sometimes be traded-off to obtain faster speeds.

For the matrix inversion algorithm, the amount of memory (double words for a complex number) needed is at most M^2. To see this, we observe from steps A-F that it is necessary to store $M(M-1)/2$ complex numbers. For steps G-M, it is necessary to store at most $M^2/2$ complex numbers. This results because it can be shown that if $\Omega_k = (\Omega_{mn}^{(k)})$, $k = 1, 2, \ldots, M$, then

$$\Omega_{mn}^{(k)} = \Omega_{(k-m+1)(k-n+1)}^{(k)*}; \quad m,n = 1, 2, \ldots, k. \tag{30}$$

Therefore, only half of the elements of the Ω_k matrix need to be stored. Since $k \leqslant M$, this number is at most $M^2/2$. Hence, it follows that the maximum memory needed for steps A-M is M^2. Storage requirements for most matrix inversion algorithms are of the order, M^2 [2]. Thus there is no advantage in eliminating memory by using the matrix inversion algorithm presented in this report.

A good indication of the speed of an algorithm is the number of multiplications (Xs) that are necessary to perform the algorithm. Multiplications and divisions that are implemented digitally are generally much slower operations than the addition, subtraction, loading, and storing operations and hence may account for the greater portion of the processing time. For steps A-F of the matrix inversion algorithm, the approximate number of Xs is $2M^3/3$ and for steps G-M the approximate number of Xs is $M^4/4$. Hence, the total number of Xs is of the order of M^4. For most direct methods of matrix inversion the number of Xs is of the order of M^3 [2]. Thus it is seen that the algorithm presented in this report is comparatively slow at least when implemented in pure software.

There are two other disadvantages associated with this matrix inversion algorithm. First, if the given hermitian Toepiltz matrix H is singular, the algorithm does not indicate this. Second, if H is

nonsingular, the intermediate UDHTMs employed in the algorithm may be singular. In this case, the algorithm fails. It is possible to determine if an intermediate UDHTM is singular by noting whether any of the values of λ_{ll}, $l = 1, 2, \ldots, 2k - 1$, calculated in Eq. (28) are zero. If any of these values are zero, then the given UDHTM is singular and the algorithm fails.

SUMMARY AND CONCLUSIONS

A new method for obtaining the inverse of a hermitian Toepiltz matrix was presented. In addition, a related technique for finding the solution of the system of linear equations, $H\bar{x} = \bar{c}$, where H is a hermitian Toepiltz matrix, was developed. Efficient algorithmic procedures for both of these methods were listed.

ACKNOWLEDGMENT

I thank Emanuel Vegh for his suggestions and various helpful discussions in the preparation of this report.

REFERENCES

1. O. Taussky and L.J. Paige (eds.), *Simultaneous Linear Equations and the Determination of Eigenvalues*, National Bureau of Standards Applied Mathematics Series No. 29, U.S. Government Printing Office, Washington, D.C., 1953.

2. J.R. Westlake, *A Handbook of Numerical Matrix Inversion and Solution of Linear Equations*, Krieger Publishing Co., New York, 1975.

3. A.S. Householder, *Principles of Numerical Analysis*, McGraw-Hill, New York, 1953.

4. L. Fox, *An Introduction to Numerical Linear Algebra*, Clarendon, Oxford, 1964.

5. R.S. Varga, *Matrix Iterative Analysis*, Prentice-Hall, Englewood Cliffs, N.J., 1962.

6. N. Wiener, *Extrapolation, Interpolation, and Smoothing of Stationary Time Series with Engineering Applications*, John Wiley and Sons, Inc., New York, 1950.

7. John P. Burg, "Maximum Entropy Spectral Analysis," Ph.D. Thesis, Stanford Univ., #75-25, 499 Univ. Microfilms Intl., Ann Arbor, Mich., 1975.

Appendix A
INVERSE OF AN UP-DOWN HERMITIAN TOEPILTZ MATRIX

In this appendix, we derive the inverse of a nonsingular up-down hermitian Toepiltz matrix (UDHTM). Let A be a $N \times N$ UDHTM such that

$$A = ((a_0, a_1, a_2, \ldots, a_{M-1}, a_{M-1}^*, a_{M-2}^*, \ldots, a_1^*)) \tag{A1}$$

where a_0 is real and $N = 2M - 1$.

The methodology of finding A^{-1} is embedded in discrete-Fourier-series (DFS) analysis. The DFS periodic convolution theorem [A1] states that if $x(k)$, $y(k)$ and $z(k)$, $k = \ldots -2, -1, 0, 1, 2, \ldots$, are periodic sequences with a period equal to N and

$$z(n) = \sum_{m=0}^{N-1} x(n)y(n-m), \tag{A2}$$

then

$$Z(k) = X(k)Y(k) \tag{A3}$$

where $X(k)$, $Y(k)$, and $Z(k)$ are the Nth order DFSs of $x(n)$, $y(n)$, and $z(n)$ respectively. Recall that a DFS is defined by the mapping of a sequence, $u(n)$, of length N into a sequence, $U(k)$, through the transformation

$$U(k) = \sum_{n=0}^{N-1} u(n)\, W_N^{kn}, \quad k = 0, 1, 2, \ldots, N-1 \tag{A4}$$

where $W_N = \exp\{-2\pi j/N\}$, $j = \sqrt{-1}$. The sequence, $u(n)$, can be found from the inverse transformation

$$u(n) = \frac{1}{N} \sum_{k=0}^{N-1} U(k)\, W_N^{-kn}, \quad n = 0, 1, 2, \ldots, N-1. \tag{A5}$$

Let us define F_N to be an $N \times N$ matrix such that

$$F_N = (f_{mn}); \; f_{mn} = W_N^{(m-1)(n-1)}; \; m,n = 1, 2, \ldots, N. \tag{A6}$$

The matrix F_N will be called the Nth order DFS matrix because we can rewrite Eq. (A4) in matrix notation as

$$\overline{U} = F_N \overline{u} \tag{A7}$$

where $\overline{U} = (U(0), U(1), \ldots, U(N-1))^T$, $\overline{u} = (u(0), u(1), \ldots, u(N-1))^T$, and T denotes transpose. The DFS matrix has the property that

$$F_N^{-1} = \frac{1}{N} F_N^*. \tag{A8}$$

This property can be shown by rewriting the inverse DFS transformation, Eq. (A5), in matrix notation and comparing this to Eq. (A7).

Let us define a periodic sequence $y(n)$, $n = 0, 1, 2, \ldots, N-1$ such that

$$y(n) = \begin{cases} a_n & n = 0, 1, \ldots, M-1 \\ a^*_{N-n} & n = M, M+1, \ldots, N-1. \end{cases} \tag{A9}$$

It can be shown that $Y(k)$, $k = 0, 1, \ldots, N-1$ are real.

We can now rewrite Eq. (A2) in matrix notation and show that $\bar{z} = A\bar{x}$ or equivalently

$$\bar{x} = A^{-1}\bar{z} \tag{A10}$$

where A is an $N \times N$ UDHTM defined by Eq. (A1), $\bar{z} = (z(0), \ldots, z(N-1))^T$, and $\bar{x} = (x(0), \ldots, x(N-1))^T$. We can also write Eq. (A3) in matrix notation as

$$\bar{Z} = \Lambda \bar{X} \tag{A11}$$

where Λ is a diagonal matrix with real diagonal elements $\lambda_{kk} = Y(k-1)$, $k = 1, \ldots, N$, $\bar{Z} = (Z(0), \ldots, Z(N-1))^T$, and $\bar{X} = (X(0), \ldots, X(N-1))^T$. However, we know that $\bar{Z} = F_N \bar{z}$ and $\bar{X} = F_N \bar{x}$, so that Eq. (A11) can be rewritten as

$$F_N \bar{z} = \Lambda F_N \bar{x}. \tag{A12}$$

If we solve for $\bar{x}$ in Eq. (A12) and use Eq. (A8), we find that

$$\bar{x} = \frac{1}{N} F_N^* \Lambda^{-1} F_N \bar{z}. \tag{A13}$$

Subtracting Eq. (A13) from Eq. (A10), we see that

$$0 = \left(\frac{1}{N} F_N^* \Lambda^{-1} F_N - A^{-1} \right) \bar{z}. \tag{A14}$$

Since $\bar{z}$ can be chosen arbitrarily, this implies that

$$A^{-1} = \frac{1}{N} F_N^* \Lambda^{-1} F_N. \tag{A15}$$

We summarize our result by the following theorem:

Theorem: If A is an $N \times N$ UDHTM, then A can be written in the form

$$A = \frac{1}{N} F_n^* \Lambda F_N$$

where F_N is the Nth order DSF matrix and Λ is an $N \times N$ diagonal matrix with real elements. In addition, if A is nonsingular, then A^{-1} has the form

$$A^{-1} = \frac{1}{N} F_N^* \Lambda^{-1} F_N.$$

We now prove the following theorem:

Theorem: If A is a nonsingular UDHTM, then A^{-1} is a UDHTM.

Proof:

Let us derive an individual element of the matrix A^{-1} by using Eq. (A15). By direct calculation, it can be shown that if $A^{-1} = (\alpha_{mn})$ $m,n = 1, 2, \ldots, N$, then

$$\alpha_{mn} = \frac{1}{N} \sum_{k=0}^{N-1} \lambda_{k+1}^{-1} W_N^{(n-m)k}. \tag{A16}$$

We show that A^{-1} is hermitian by using the fact that λ_k, $k = 1, \ldots N$ is real and $W_N^{-1} = W_N^*$. Thus

$$\begin{aligned}
\alpha_{nm}^* &= \left[\frac{1}{N} \sum_{k=0}^{N-1} \lambda_{k+1}^{-1} W_N^{(m-n)k}\right]^* \\
&= \frac{1}{N} \sum_{k=0}^{N-1} \lambda_{k+1}^{-1} \left[W_N^{(m-n)k}\right]^* \\
&= \frac{1}{N} \sum_{k=0}^{N-1} \lambda_{k+1}^{-1} W_N^{(n-m)k} \\
&= \alpha_{mn}.
\end{aligned}$$

Also, it is readily shown from Eq. (A16) that the diagonal elements $(m=n)$ are real.

We use the form of Eq. (A16) to show that A^{-1} is Toepiltz. We see that it is possible to write α_{mn} in the form $\alpha_{mn} = \beta_{m-n}$ for all m and n, which is exactly the form of a Toepiltz matrix.

We show that A^{-1} has the up-down property by demonstrating that for the elements in the first row that

$$\alpha_{1n} = \alpha_{1(N-n+2)}^*. \tag{A17}$$

We do this as follows:

$$\begin{aligned}
\alpha_{1(N-n+2)}^* &= \left[\frac{1}{N} \sum_{k=0}^{N-1} \lambda_{k+1}^{-1} W_N^{(N-n+1)}\right]^* \\
&= \frac{1}{N} \sum_{k=0}^{N-1} \lambda_{k+1}^{-1} \left[W_N^{(N-n+1)k}\right]^* \\
&= \frac{1}{N} \sum_{k=0}^{N-1} \lambda_{k+1}^{-1} W_N^{(n-1)k} \\
&= \alpha_{1n}.
\end{aligned}$$

Hence, the theorem is proved.

We see from Eq. (A16) that in order to find the elements of the first row of A^{-1}, we can write

$$\alpha_{1n} = \frac{1}{N} \sum_{k=0}^{N-1} \lambda_{k+1}^{-1} W_N^{(n-1)k}. \tag{A18}$$

However, we notice that the form of Eq. (A18) is that of a DFS (see Eq. (A4)) except for a scalar factor of $1/N$. Hence, to generate the first row of A^{-1}, we merely find the Nth order DFS of the sequence $\lambda_1^{-1}, \lambda_2^{-1}, \ldots, \lambda_N^{-1}$ and divide all elements of the DFS by N. Therefore, since the first row of a hermitian Toepiltz matrix specifies the entire matrix, we have found a simple method of generating the

inverse of A^{-1}. Firstly, we generate $\lambda_1, \lambda_2, \ldots, \lambda_N$ by calculating the Nth order DFS of the sequence $a_0, a_1, \ldots, a_{M-1}, a_{M-2}^*, \ldots, a_1^*$. Secondly, the first row of A^{-1} is found by calculating the Nth order DFS of the sequence $\lambda_1^{-1}, \lambda_2^{-1}, \ldots, \lambda_N^{-1}$ and then dividing all elements of the DFS by N. Finally, because A^{-1} is a hermitian Toepiltz matrix, all other elements of the matrix are specified by elements of the first row.

REFERENCES

A1. A.V. Oppenheim and R.W. Schafer, *Digital Signal Processing*, Prentice-Hall, Inc., Englewood Cliffs, N.J., 1975.

Appendix B
PROGRAM LISTING OF THE MATRIX INVERSION ALGORITHM

```
      SUBROUTINE TOEPLZ(H,OMEGA,M,N)
C     ****************************************************
C
C     THIS SUBROUTINE FINDS THE INVERSE OF A GIVEN NONSINGULAR
C     HERMITIAN TOEPLITZ MATRIX WHERE
C
C         H=THE HERMITIAN TOEPLITZ MATRIX
C           (NOTE THE DIAGONAL ELEMENTS MUST BE REAL)
C
C     OMEGA=THE MATRIX INVERSE OF H
C
C         M=THE ROW OR COLUMN DIMENSION OF H
C
C         N=2M-1
C
C     THE ALGORITHM MAY FAIL IF AN INTERMEDIATE MATRIX THAT IS
C     USED IN CALCULATING THE INVERSE IS SINGULAR. IF THIS
C     OCURRS THE MESSAGE "ALGORITHM FAILS" IS PRINTED.
C     *****************************************************
      IMPLICIT COMPLEX*8 (B-H,O,Q-Z)
      DIMENSION H(M,M),T(N,M),F(N,N),TP(N,M),OMEGA(M,M),
     1 Q(N,N),Q11(M,M),Q12(M,M),Q21(M,M),AAO(M,M),B(N,1),
     1 THETA(N,N)
      DATA PI/3.1415929/,AERROR/.000001/
C     INITIALIZE MATRIX CONSTANTS
C     INITIALIZE T MATRIX
      DO 300 K=1,M
      T(K,M)=H(K,1)
  300 CONTINUE
      M1=M+1
      DO 400 K=M1,N
      T(K,M)=CONJG(H(N+2-K,1))
  400 CONTINUE
      M1=M-1
C     FIND SUCCESSIVE DFS
      DO 500 MMM=1,M1
      MM=M+1-MMM
      NN=2.*MM-1
C     COMPUTE DFS MATRIX OF ORDER NN
      A1=-2.*PI/NN
      DO 600 K=1,NN
      DO 700 L=1,NN
      AC=COS((K-1)*(L-1)*A1)
      AS=SIN((K-1)*(L-1)*A1)
      F(K,L)=CMPLX(AC,AS)
```

```
  700 CONTINUE
  600 CONTINUE
      DO 3000 K=1,NN
 3000 TP(K,1)=T(K,MM)
C     FIND DFS OF TP
      DO 3200 K=1,NN
      F(K,1)=CMPLX(0.,0.)
      DO 3300 L=1,NN
 3300 F(K,1)=F(K,1)+P(K,L)*TP(L,1)
 3200 CONTINUE
      DO 800 K=1,NN
        F(K,1)=REAL(F(K,1))
      F(K,1)=1./(NN*F(K,1))
  800 CONTINUE
      DO 3400 K=1,NN
      IP(K,1)=CMPLX(0.,0.)
      DO 3500 L=1,NN
 3500 IP(K,1)=IP(K,1)+F(K,L)*A(L,1)
        A=ABS(REAL(IP(K,1)))
        IF(A.LT.AERROR)TYPE 100
  100   FORMAT(1X,'ALGORITHM FAILS')
        IF(A.LT.AERROR)RETURN
 3400 CONTINUE
        IP(1,1)=REAL(TP(1,1))
        I(1,MM)=TP(1,1)
      DO 900 K=2,NN
      IP(K,1)=CONJG(TP(K,1))
      I(K,MM)=TP(K,1)
  900 CONTINUE
      MM1=MM-1
      DO 1000 K=1,MM1
      I(K,MM1)=TP(K,1)
 1000 CONTINUE
      NN1=2.*MM1-1
      DO 1100 K=MM,NN1
      I(K,MM1)=CONJG(TP(NN1+2-K,1))
 1100 CONTINUE
  500 CONTINUE
C COMPUTE INVERSE MATRIX
      OMEGA(1,1)=1./T(1,2)
      DO 1200 MM=2,M
      NN=2.*MM-1
C   FIND Q MATRIX
        Q(1,1)=I(1,MM)
      DO 1300 K=2,NN
 1300   Q(1,K)=CONJG(T(K,MM))
      DO 1400 K=2,NN
 1400 Q(K,1)=CONJG(Q(1,K))
      DO 1500 J=2,NN
      DO 1600 I=2,NN
      Q(I,J)=Q(I-1,J-1)
 1600 CONTINUE
 1500 CONTINUE
```

```
C   FIND Q11,Q12,Q21
      MM1=MM-1
      MMP1=MM+1
      DO 3600 K=1,MM
      DO 3700 L=1,MM
 3700 Q11(K,L)=Q(K,L)
 3600 CONTINUE
      DO 3800 K=1,MM
      DO 3900 L=1,MM1
 3900 Q12(K,L)=Q(K,MM+L)
 3800 CONTINUE
      DO 4000 K=1,MM1
      DO 4100 L=1,MM
 4100 Q21(K,L)=Q(MM+K,L)
 4000 CONTINUE
C   COMPUTE INVERSE
      DO 4200 I=1,MM1
      DO 4300 J=1,MM
      RHO(I,J)=CMPLX(0.,0.)
      DO 4400 K=1,MM1
 4400 RHO(I,J)=RHO(I,J)+OMEGA(I,K)*Q21(K,J)
 4300 CONTINUE
 4200 CONTINUE
      DO 4500 I=1,MM
      DO 4600 J=1,MM
      THETA(I,J)=CMPLX(0.,0.)
      DO 4700 K=1,MM1
 4700 THETA(I,J)=THETA(I,J)+Q12(I,K)*RHO(K,J)
 4600 CONTINUE
 4500 CONTINUE
      DO 4800 I=1,MM
      DO 4900 J=1,MM
 4900 OMEGA(I,J)=Q11(I,J)-THETA(I,J)
 4800 CONTINUE
 1200 CONTINUE
        RETURN
      END
```


FILMED
8